

Ambiente y proyecto inicial

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas a este nuevo video dentro del curso de Servlet. En este caso vamos a ver las configuraciones que necesitamos tener en nuestras máquinas, los programas que vamos a necesitar tener para poder ya trabajar en servlets.

[00:18] Entonces para aquellos que cayeron acá de paracaídas directamente en este curso vamos a explicar todo lo que necesitan tener. Aquellos que ya hayan hecho el curso anterior, ya pueden ir directamente al siguiente video porque no vamos a ver nada nuevo, solo vamos a explicar las cosas que necesitamos tener dentro de nuestras máquinas.

[00:36] Bien, entonces comenzamos con Java. Vamos a necesitar Java en su versión 17, que es la que yo voy a estar usando. Les recomiendo mucho tener esa misma versión que yo, igual que en todas las otras herramientas que tengan la misma versión que yo, porque eso les va a permitir que si yo tengo algún tipo de error, posiblemente ustedes tengan los mismos errores y sepan cómo solucionarlos, porque yo les voy a decir cómo solucionarlos, si es que ocurre.

[01:02] Si tienen otras versiones diferentes puede ser que tengan errores diferentes o problemas diferentes que no van a estar tal vez en este video. Entonces por eso es mejor tener las mismas versiones que yo voy a utilizar. Entonces, si ustedes ya tienen Java instalado, pueden utilizar la versión que ustedes quieran desde la versión 8 en adelante, pero como les dije, les recomiendo tener la versión 17.

[01:23] Si no tienen ya instalado, no necesitan bajar la versión 17, ya que nuestra siguiente herramienta, que es Eclipse, ya trae un instalador también de Java dentro, entonces ahí, dentro de la instalación de Eclipse, ustedes van a poder automáticamente instalar Java 17 también.

[01:40] Para bajar Eclipse es ir a download aquí y pueden utilizar cualquier versión que aparezca acá. Son todas muy muy parecidas. Y finalmente vamos a utilizar Tomcat en su versión 10.0. En este caso aparece la 10.0.22 pero ustedes pueden usar la cualquier versión que aparezca siempre y cuando sea 10.0, porque ya con Tomcat en sus versiones superiores, posiblemente haya mucha variación en problemas o cosas que puedan ocurrir. Entonces, mejor tener la 10.0.

[02:17] Para bajar Tomcat en su versión 10.0, tienes que hacer clic en este .zip que aparece acá, lo bajan y lo descomprimen en alguna carpeta que ustedes quieran. Sobre la configuración de Eclipse me olvide decir que tienen que no tienen que hacer un siguiente, siguiente, siguiente, siguiente ya que eso va a provocar que instalen la versión más simple de Eclipse y nosotros en realidad necesitamos una versión especial que es la versión llamada Enterprise Java and web developers.

[02:52] Entonces instalen esa versión dentro de las opciones que les va a aparecer ahí de instalación junto con Java 17, también como había dicho. Perfecto. Una vez que ya han bajado todas las herramientas Eclipse, Apache e instalador Java, vamos a configurar Eclipse.

[03:09] Al comienzo puede ser que les aparezca una ventana de bienvenido y cosas así, es simplemente una pestaña que ustedes pueden cerrarla con una x. Luego de hacer eso, van a ver algo parecido con esta pantalla. Acá esta parte de la derecha yo normalmente la minimizo, no la estamos utilizando.

[03:28] Si no ven una pantalla parecida con esto puede ser que no tengan clickeable esta versión, esta parte de acá que dice Java EE. Ustedes

simplemente clican ahí y se les va a aparecer el layout o la distribución de los elementos como tengo yo acá. Puede ser que a ustedes les aparezca así más o menos. Simplemente clicando acá en ese icono obtienen esta distribución.

[03:56] Siguiente. Esta pestaña de servers que aparece acá es muy importante para nosotros. Si ustedes no la ven, por ejemplo yo voy a cerrarla acá para simular que no está, pueden ir a la lupita que aparece ahí arriba, escribir bueno, acá me apareció server, pero ustedes pueden escribir servers, y es ese icono y ya tienen esto que vamos a utilizar nosotros.

[04:20] Acá es donde nosotros agregamos nuestros servers, que en este caso nosotros vamos a utilizar Tomcat, entonces hacen clic ahí en ese click this link to create new server. Y acá en esta lista yo voy a aumentarla para que se vea mejor, vamos a ir a la carpeta apache, abrimos esa carpeta y como les dije yo tengo la versión 10.0. Entonces vamos a elegir esa versión.

[04:47] Acá vamos a hacer un siguiente. Y acá en esta parte de acá tenemos que decir dónde está nuestro zip que nosotros hemos ya descomprimido, nosotros ya extrajimos los datos en alguna carpeta, esa carpeta es la que tienen que colocar acá. Entonces yo la tengo en esa dirección, si ustedes la tienen en otra dirección van a browse y acá ustedes seleccionan la dirección. Bien, pueden buscarla.

[05:14] Y finalmente acá en JRE ustedes acá es donde seleccionan la opción de Java que ustedes tienen instalada, yo voy a colocar workbench default JRE, que es la versión que yo instalé, también junto con Eclipse, entonces yo instalé Java a través de Eclipse, entonces usan esa opción si ustedes tienen Java instalado por otro lado, posiblemente les aparezca en esa lista y pueden seleccionar esa opción. Bien.

[05:42] Next y finish. Acá tenemos dentro nuestro Tomcat bien configurado y tenemos del lado izquierdo una carpeta que es un proyecto llamado Servers. Esa carpeta tiene todas las configuraciones de Tomcat 10. Entonces ustedes no

tienen que cerrarla, o sea, no hagan clic derecho, delete ni close project. Ninguno de los dos, porque si hacen eso Tomcat deja de funcionar.

[06:08] Entonces vamos a tener eso y lo que tenemos que hacer ahora es importar nuestro proyecto. Yo les dejé un .zip ahí, dentro de los archivos del curso, entonces vamos a importarlo. Tienen en este caso yo tengo mi .zip dentro de mis descargas, acá tengo mi .zip, entonces ustedes tienen que hacer un file, clic en file y acá colocan en import. Dentro de import en la carpeta de general tienen existing projects into workspace.

[06:42] Vamos a importar un proyecto que ya existe, ya iniciado, entonces next, habiendo seleccionado esa opción, next. Y acá tienen que hacer clic en ese select arcade, vamos a seleccionar un archivo y ponemos browse acá. Nosotros seleccionamos el archivo que yo tengo entonces ese es el archivo que yo les he dejado doble clic en él.

[07:10] Asegúrense que está tildada esa opción y simplemente hacemos un finish. Bien, perfecto. Tenemos acá arriba a la izquierda nuestro proyecto. Ahora puede ser que ocurran dos situaciones. Uno, que ustedes puedan automáticamente enviar el proyecto acá para Tomcat, arrastrándolo hacia nuestro Tomcat en la parte de acá.

[07:32] Puede ser que ustedes ya puedan soltarlo y funcione o puede ser que ocurra como me está pasando a mí, que yo al intentar llevarlo aparece un símbolo de que no es posible llevar ese proyecto a Tomcat. Caso les ocurra eso que me aparece acá, pueden hacer clic derecho en gerenciador, propiedades y acá dentro de Project facets, no sé si se pronuncia así, pero yo lo pronuncio así

[07:59] Vamos a dynamic web model, tienen que tildar esta opción y en versión colocan la versión 5.0. Además de esa opción tienen que tildar la opción de Java que yo tengo la versión 17. Si ustedes tienen otra versión colocan otra versión y en JavaScript, hecho eso van a hacer un apply.

[08:22] Clic en apply, no hagan apply and close, solo apply, y después vamos a target runtime, esa opción que tenemos acá y van a tildar la opción de Apache Tomcat. Una vez hecho eso, apply and close. Y con esto, en teoría, ya deberían poder enviar gerenciador al top. Vean que yo puedo ya arrastrarlo. Me deja hacer eso.

[08:45] Hay otra opción en vez de arrastrarlo, pueden hacer clic derecho propiedades, no. Perdón, clic derecho, add and remove, y acá les va a aparecer el proyecto del lado izquierdo y hacen clic en él y pueden hacer clic en ADV y ya va a aparecer del lado derecho y significa que ya lo estarían agregando a ese proyecto a nuestro servidor Tomcat.

[09:10] Ahora hemos iniciado nuestro proyecto con esto, yo puedo seleccionar Tomcat y hacer clic en el botoncito verde de start server. Con esto va a iniciar el servidor, y si nosotros nos vamos a la consola acá nos aparece si por ejemplo si apareció algún error, va a aparecer acá en la consola, entonces siempre vean esa consola a ver si hay algún problema.

[09:34] Una vez hecho eso, acá no aparece ningún problema, entonces vamos a ir a Google Chrome y vamos a probar nuestro proyecto. Vamos a colocar localhost:8080/gerenciador, perdón que estuve probando unas cosas, /gerenciador, enter y vean que aparece bienvenido al curso de servlets de Alura.

[09:52] Entonces si yo me voy a listaEmpresas. No. ListaEmpresas normal. Ahí está, listaEmpresas. Ahí podemos ver nuestras listas de empresas y podemos hacer clic en modificar, modificar una empresa, acá 2021, enviar, perfecto y aparece, está funcionando todo. Perfecto. Ya tenemos nuestro proyecto funcionando.

[10:16] Ahora me gustaría explicar cuáles son las carpetas y distribución que tenemos dentro de nuestro archivos. Dentro de gerenciador, si ustedes abren el proyecto gerenciador y van a source, main webapp. Acá dentro es donde

tenemos la mayoría de las cosas que están relacionadas a las views o archivos que nuestros usuarios van a ver.

[10:39] Entonces tenemos, por ejemplo un `bienvenido.html` y tenemos varios `jsp`. Vamos a hacer clic acá en `formNuevaEmpresa`, por ejemplo, y tenemos algunas cosas interesantes. Acá este `jsp` es muy parecido a un `HTML`, solo que con algunas diferencias, algunos agregados. Este `jsp` por ejemplo tiene `expression language`, que es esta forma de colocar una variable dentro de nuestro `HTML`.

[11:07] Bien. Y también nosotros tenemos nuestros `jstl`, que nos permite agregar algunas tags, algunas etiquetas que nos dan funcionalidades nuevas, por ejemplo en este caso, nosotros podemos limpiar una URL con una de esas taglibs, con esta taglib. Entonces tenemos varias opciones y varias herramientas que utilizar con esas taglibs.

[11:43] Por ejemplo, podemos formatear una fecha acá utilizando un objeto de tipo fecha, nosotros podemos colocar un patrón y nos devuelve en el `HTML` esa fecha con ese patrón. Bien, perfecto. Dentro de `webapp` también tenemos otra carpeta muy importante `WEB-INF` que esas `WEB-INF` acá nosotros creamos una carpeta llamada `Lib` que es donde tenemos nuestras librerías, que acá es donde están, por ejemplo nuestras `jstl` que nos permiten agregar todas esas tags con formatación, etcétera.

[12:18] Entonces acá es donde están todas esas tags. Bien, también tenemos un archivo muy, muy importante llamado `web.xml`, que en él es donde tenemos nuestro mapeo de servlet con URL. Ya vamos a ver enseguida qué es eso, pero es un archivo muy, muy importante, donde acá podemos configurar varias cosas de nuestro proyecto.

[12:42] Luego voy a cerrar acá “`Ctrl + Shift + W`”, cierro todas las pestañitas y si nos vamos a `src/main/java` acá tenemos un paquete llamado `com.alura.genciador.servlet`. En ese paquete, nosotros tenemos nuestros

modelos y nuestros servlet. Entonces dentro de nuestros modelos nosotros tenemos este DB, por ejemplo, que es una simulación de base de datos.

[13:05] No es una base de datos real, sino que es un archivo que reemplazaría de una forma muy precaria a una base de datos de verdad, pero para nuestro proyecto sirve, ya que cumple su función de poder guardar cosas y obtener datos de ese archivo. En este archivo DB que es nuestra base de datos simulada tenemos una lista de empresas.

[13:30] En esa lista de empresas nosotros inicialmente creamos empresas ya al momento de inicializar el servidor, ya creamos dos empresas, Alura y Caelum, y las agregamos a esa lista. Además de crear las empresas y tenerlas en la lista, podemos agregar una empresa, podemos obtener una empresa, eliminar una empresa y buscar una empresa. Entonces funciona como si fuera una base de datos.

[14:00] Además de la base de datos, dentro de lo que es nuestro modelo, tenemos también nuestras empresas. Esas empresas tienen un nombre que es un string y una fecha, que es un tipo date. Bien. Además de eso, tenemos un id que gracias a nuestra base de datos es autoincremental y secuencial. Y entonces esa empresa tiene los métodos de get y set de cada uno de sus atributos.

[14:25] Y bien, es eso, básicamente empresa es eso. Finalmente nosotros tenemos nuestros servlets, voy a hacer “Ctrl + Shift + W” para cerrar los archivos y nosotros tenemos, por ejemplo, nuestra nueva empresa servlet, que es un buen ejemplo para mostrar qué es un servlet.

[14:43] Un Servlet es un objeto Java al cual es posible acceder a él utilizando el protocolo http. Entonces, a través de una URL yo puedo acceder a ese objeto Java, entonces nosotros para considerar una clase como que sea un servlet para que sea un servlet, nosotros tenemos que hacer un extends de http servlet.

[15:05] Con eso, entonces ya Tomcat ya sabe que eso es un servlet. Bien, dentro de ese servlet, nosotros tenemos un método, en este caso un doPost, pero ese método puede ser un doGet o un service. ¿Qué son estos métodos? Estos métodos nos permiten atender a una petición http. En este caso estamos atendiendo peticiones http post, pero podemos atender peticiones get. Podemos tener otras peticiones put, libertad, etcétera, pero en este caso utilizamos doPost, doGet y el service.

[15:41] ¿Qué es el service? El service atiende tanto http, perdón, tanto peticiones get como post, las dos. En este doPost, en esa función tenemos un método y él recibe dos parámetros, un request y un response, el request es lo que nosotros recibimos del navegador del cliente y el response es lo que nosotros vamos a enviar como respuesta al cliente.

[16:12] Dentro de este doPost, si nosotros obtenemos unos parámetros que el cliente puede enviarnos, en este caso por ejemplo es un nombre y una fecha. Esos parámetros, siempre que nosotros obtenemos los obtenemos son strings. Entonces en algunos casos es mejor hacer un cast. ¿Qué es un cast? Nosotros transformamos, por ejemplo, en este caso una fecha de string para el tipo date.

[16:40] Esto de acá es un cast. ¿Por qué hacemos eso? Porque a veces es mucho más fácil trabajar con un dato de forma más nativa, digamos una fecha es fácil de trabajarla como date, ya que tenemos varios métodos dentro del tipo date, que nos ayudan a formatear y hacer cosas.

[17:01] Por ejemplo, es SimpleDateFormat nos permite obtener la fecha y en ese formato. Después de eso, después de hacer el parsing, nosotros tenemos acá la creación de una empresa, en este caso, entonces vamos a crear los objetos que van a ir después a la base de datos y finalmente tenemos nuestra lógica, que en este caso nosotros instanciamos la base de datos y agregamos una empresa en la base de datos.

[17:30] Este servlet sirve para eso y además de eso, nosotros lo que hacemos es agregar un atributo a nuestro request llamado empresa, que es con el nombre de nuestra empresa que hemos acabamos de crear. Y en este caso estamos haciendo un redirect, le estamos diciendo al navegador que después pida de nuevo la lista de empresas.

[17:54] Podríamos hacer un redirect como podríamos hacer un requestDispatcher, por ejemplo también. En síntesis, eso es lo que nosotros tenemos en la mayoría de nuestros servlets, siempre la misma estructura, obtener parámetros, casts cuando es necesario, crear empresas o hacer alguna cosa con nuestra base de datos y enviar un request y un response. Bien.

[18:17] Una cosa que me olvidé de mencionar es que acá podemos tener anotaciones. Generalmente dentro de los servlets podemos utilizar la notación webServlet, con "Ctrl + espacio" ustedes autocompletan y hacen el import automáticamente. Donde acá se coloca una URL, por ejemplo barra nuevaEmpresa.

[18:39] Con esto nosotros lo que crearíamos es crear una URL en la cual la persona puede llamar, y llega a este servlet. Pero en este caso no tengo ese webServlet en esta clase. Voy a hacer un "Ctrl + Z" para eliminar también los imports, no tenemos ese webServlet acá porque nosotros tenemos configurada esa URL dentro de nuestro web.xml.

[19:07] Y acá aparece nuestra nueva empresaServlet. En ese web.xml nosotros tenemos un displayName, por ejemplo, que es nuevaEmpresaServlet y un servletName. Y esta parte de acá nos está vinculando a ese ServletName que es nuevaEmpresaServlet con una clase, que es nuestra nueva empresaServlet.

[19:29] Y además de eso acá nosotros vinculamos ese ServletName con una URL. Bien. Entonces esa anotación que habría puesto acá como webServlet, la tendríamos acá dentro de nuestro web.xml. Pero ya van a ver en el futuro más

adelante vamos a ver que ya no vamos a utilizar tanto este web.xml y s
vamos a utilizar un poco más de anotaciones, bien.